NUCLEONICA – SOFTWARE DESIGN PATTERNS

C. GRAMMES, C. LAMBERT, H. FRIES
DIaLOGIKa GmbH, Pascalschacht 1, D-66125 Saarbrücken – Germany
Dokumenta S.A., 16, rue d'Epernay, L-1490 Luxembourg – Luxembourg

J. MAGILL
European Commission, Joint Research Centre,
Institute for Transuranium Elements, Postfach 2340, 76125 Karlsruhe, Germany

ABSTRACT

NUCLEONICA is the latest development in a family of information systems for the nuclear science community. Throughout this contribution some aspects of the internal structure of NUCLEONICA are described, in order to show the flexibility of the software architecture. In a typical 3-tier architecture it combines efficient databases with modern rich internet presentation techniques and a modular structure. Based on the Microsoft .NET framework it is well-suited to provide the ease of use today's computer users have come to expect. It is shown how web server, webservices, application libraries and the database repository are smoothly integrating with third party applications like MediaWiki that is used as a general help facility and as an interactive Q&A discussion forum.

1. Introduction

The evolution from the first predecessor through to NUCLEONICA reflects the software technical paradigm change from fat clients to a modern web application using latest Web 2.0 technology. NUCLEONICA's modular structure with the idea of *Software as a Service* in mind is well suited for integrating newly developed application modules such as the radiological dispersion module as well as well-known legacy codes such as KORIGEN [1] for nuclide depletion calculations in nuclear reactors. It was even possible to integrate an open-source framework like Mediawiki [2], being built on Apache, PHP and a mySQL database. Despite totally different code bases and database requirements a common single-sign-on for all applications could be realized.

The idea of providing an electronic knowledge base of physical nuclide data has a long history. There are two predecessor applications of NUCLEONICA each using a unique software technical approach – due to the technical possibilities and habits existing at the time, Nuclides 2000 [3] was a fat client Visual Basic application that installed on Windows PC's and was distributed on CD-ROM. It consisted of a database of nuclear data (Microsoft Access) and several applications, e.g. Nuclide Explorer (with far less functionality than in NUCLEONICA) and a decay engine. In that approach, updates weren't easily possible (new CD-ROMs had to be produced) and the platform was restricted to Windows based PCs. All algorithms were frozen to the scientific state when the CD-ROM was produced.

The next step towards web technology came with Nuclides.net in 2003 [4]. Still produced on CD-ROM with similar a design approach as that of Nuclides 2000, it shifted the applications part such as Decay Engine to a web application server at ITU in Karlsruhe, thus providing the possibility to keep step with the improving scientific models.

NUCLEONICA [5] now broke completely with the CD-ROM based approach through being accessible completely from the World Wide Web. Since there is no need to install software,, the applications open up also to Mac and UNIX users. The database information is always up

to date and there's no longer need to support users with installation problems or to distribute updates in case of program flaws. A major challenge however was the need to provide the necessary performance, because all concurrent users share the same central systems.

2. Modular architecture based on .NET

NUCLEONICA consists of four layers. The website itself is an ASP.NET 2.0 web application. It uses an SQL Server 2005 database, several DLLs (green) and services (red) that run on the same machine.

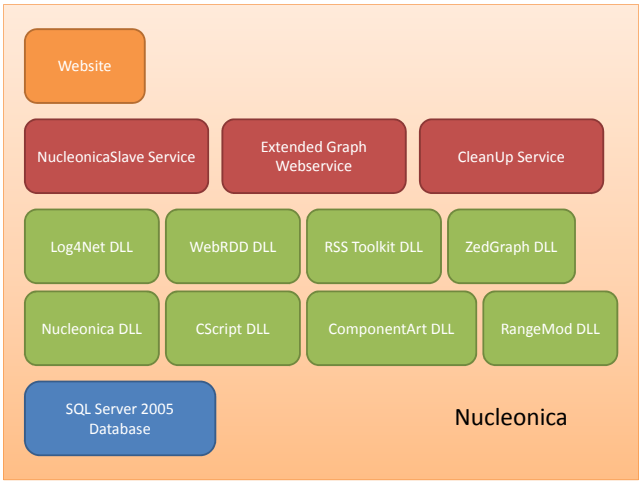

Fig. 1: NUCLEONICA Modular Architecture

Website	The ASP.NET pages the NUCLEONICA website consists of.
NucleonicaSlave	Long running asynchronous background executions
Extended Graph Webservice	Customizable generation of graphs, wraps ZedGraph.DLL
CleanUp Service	Housekeeping of temporary files
Log4Net DLL	Library for writing to the Windows event log (Error Handling)
WebRDD DLL	Library for computations in the RDD module
RSS Toolkit DLL	Library for RSS syndication
ZedGraph DLL	Library for graph generation (e.g. common graphics control)
Nucleonica DLL	Library for computations in most application pages. Common functionality like database access and collection types.
CScript DLL	Library for the NUCLEONICA Scripting Language components
ComponentArt DLL	3rd party library with enhanced user interface controls
RangeMod DLL	Library for Range & Stopping Power application
SQL Server 2005	Database server

Tab 1: NUCLEONICA modules and functionalities

The NUCLEONICA website shares a common look and feel on almost every web page. This is implemented via separate files, such as for a header and footer. Master pages are flexible page-templates allowing to "skin" and control the layout of the entire website by modifying a single template. This "visual inheritance" reduces maintenance and the overall complexity of the entire website.

NUCLEONICA uses a set of third party AJAX controls by ComponentArt [6], a vendor specializing in the creation of user interface and data visualization software for Microsoft's .NET platform. AJAX (**A**synchronous **J**avaScript **a**nd **X**ML) is a way of including content in a web page in which JavaScript code in the web page fetches some data from a server and displays it without re-fetching the entire page. The data fetched is often in XML format.

The Microsoft .NET Framework uses a managed code programming model. It consists of two main components, the common language runtime (CLR) and the .NET Framework class library. The evolution process from Nuclides 2000 to NUCLEONICA includes a variety of grown, highly complex, nuclear scientific class libraries and applications. These had to be made accessible for the fully web-based NUCLEONICA. The challenge of integrating libraries written in various programming languages into NUCLEONICA while ensuring the security, robustness and stability of the legacy code reinforced the decision for the .NET Framework. Another essential point was the cross-language compatibility of Microsoft .NET. A multitude of developers contribute to the steady evolving progress of NUCLEONICA, each of them using the programming environment best suited to his task.

3. Web 2.0 – a gimmick or a vision?

Web 2.0, far from being merely a buzz-word of modern information technology, comprises some highly interactive, individually tailorable design patterns that aim at providing an interlinked computing platform by making use of the internet not only as an information resource but also as a scientific networking platform. *"…Web 2.0 is of course a piece of jargon, nobody even knows what it means*". Tim O'Reilly, known as the founder of O'Reilly Media, coined the term Web 2.0 in 2004 encompassing a new generation of web based services – such as wikis, communication, social networking and collaborative tagging. Seeing the internet as a participation platform he understands Web 2.0 as a new way of collaborating, collecting, constructing and experiencing in the World Wide Web. The constantly emerging techniques and features of modern web applications range from content syndication to interactive rich user interfaces. The evolution from Nuclides.net to NUCLEONICA, a fully web based portal service, opened the chance to involve the user more into the application. The following approaches are realized in NUCLEONICA:

Feature / Technique♣	Example
Rich Internet Application	- Nuclide Selector - Nuclide Explorer - Alerting Service - NUCLEONICA Portal
Content Syndication and Collaboration	- NUCLEONICA [Wiki] - ITU Nuclear News - NUCLEONICA Hot Topics - Conference Calendar
Social Networking	- NUCLEONICA Community Portal

Tab. 2: Web 2.0 Features of NUCLEONICA

The main goal was to combine rich and user-friendly interfaces with the approach of a web based scientific application suite, as well as an embedded social network for nuclear science experts.

Rich Internet Application

The combination of AJAX driven controls, customizable services and user-friendly user interfaces grants a highly flexible browser-based application. One of the frequently used controls - the "Nuclide Selector" - offers an AJAX-based way of selecting the nuclides for further processing.

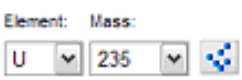

Fig 2: Nuclide Selector Control

In the first step the element is selected. This causes an AJAX *callback* to retrieve the mass numbers according to the element directly from the database. The selection of the desired mass number then initializes the *postback* to the server. Another usage of AJAX (on the server side) is the alerting control. This control notifies the logged in user about finished calculations, recent postings and messages. In these examples AJAX technology inconspicuously enhances the user interface without overloading and complicating it.

Content Syndication and Collaboration

The modern internet user not only acts as a consumer, but also as a contributor and/or producer. Therefore it was crucial to be aware of the relevance of user generated content (UGC). NUCLEONICA meets these requirements by offering the NUCLEONICA [Wiki] and the Conference Calendar. Users can exchange knowledge, information, support and events, resulting in a constantly growing mine of information – a benefit to all users. Beyond that NUCLEONICA provides a wide variety of nuclear news that can be accessed from the application or via the ITU Nuclear News Feed. The ITU nuclear news and the NUCLEONICA Hot Topics are distributed via Real Simple Syndication (RSS). RSS allows a user to subscribe to a page, not just to be linked to it. So, one is notified every time the pages changes or news are added. This makes NUCLEONICA a “live” application.

Social Networking

Another part of NUCLEONICA is the social network. Inside NUCLEONICA users can link and communicate with each other based on their common interests or scientific experiences. The network enables users to keep track of their contacts, groups and events.

4. Software as a service

Performance being an important issue with server based computing, it was necessary to have NUCLEONICA as modular as possible. The design goal bears in mind the possibility to distribute different parts of the application to different servers for load sharing and high availability issues.
The database (Microsoft SQL Server) runs totally independently and can easily be moved to another server without any code change due to the use of the .NET database interface.

Images such as graphs produced by the particular applications (Decay engine, Dosimetry and Shielding, etc.) have to be rendered in formats suitable to web browsers on one side. On the other hand it was also a design goal to support scientists with graphs of a quality appropriate for further uses such as in presentations or publications. For this purpose NUCLEONICA developers chose the freeware package ZedGraph [7] that was natively designed for Microsoft .NET and turned out to be extremely fast and produces graphs of convenient quality. It fits smoothly into the interfacing model of multi-instance applications without user interface that are typical running on a back-end server. As a consequence NUCLEONICA offers the ability to produce nice graphics not only as part of the nuclear applications but as a service for any type of data through the Extended Graph module. Moreover, in a next step this service will be offered as a webservice enabling scientists to make use of NUCLEONICA as part of their own local applications. As a proof of concept there’s already a plug-in for Microsoft Excel calling NUCLEONICA webservice for generating graphs (see Fig. 3). In future versions it is planned to provide more applications through webservice interfaces.

Long-running applications are tedious to the user especially in a web environment where one tends to consider everything that lasts longer than a few seconds as ”stuck“. The possibility of running complex computations asynchronously in the background and informing the user about results through email alerts underlines this service character of NUCLEONICA.

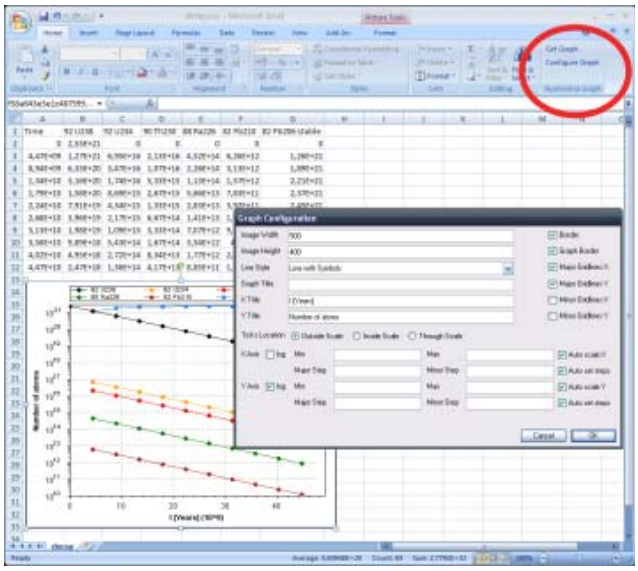

Fig. 3: Excel Plug-In for NUCLEONICA

5. Legacy Applications – don't re-invent the wheel!

Implementing NUCLEONICA as a web application was a major issue that should be made easier by re-using as much existing code as possible – both as far as system modules and libraries are concerned, and on the other hand as for nuclear and physical application modules.

As an example NUCLEONICA [Wiki] relies on the existing and well-proven technology of MediaWiki (itself built on the components of the freeware meta-package XAMPP Lite [8]). However extensions were implemented to realise for example the Single-Sign-On capability between NUCLEONICA (a .NET application) and MediaWiki, a PHP application. The NUCLEONICA user profits by not having to log in to the Wiki when coming from NUCLEONICA. Another example is the ZedGraph package (see above).

Scientific legacy code is built into the integrated nuclear software models. The webKorigen module for example is a piece of software developed over many decades. This module was integrated and combined with a modern web user interface.

Keeping in mind flexibility NUCLEONICA users are encouraged to integrate their own legacy algorithms. The integrated scripting language NUCLEONICA script, a dynamically parsed C like programming language offers access to all necessary database fields of the integrated nuclear knowledge base. It comprises all necessary language elements to write own code making use of the data and also of the integrated applications up to presenting textual and graphical results.

References

[1] J. Magill, NUCLEONICA: A Web Portal for the Nuclear Sciences, this conference.
[2] http://www.mediawiki.org
[3] J. Magill, Nuclides 2000: An Electronic Chart of the Nuclides. EUR 18737 EN (1999)
[4] J. Magill, Nuclides.net, Springer, Berlin (2003)
[5] http://www.nucleonica.net
[6] http://www.componentart.com/webui/
[7] http://zedgraph.org/
[8] http://www.apachefriends.org/en/xampp.html